

Scriptorium による電子校訂法//Edition électronique par Scriptorium

小栗栖 等
Hitoshi Ogurisu

19 janvier 2011

Scriptorium est un logiciel qui facilite l'édition des chansons de geste. Il est né et en train d'être élaboré au cours de notre édition de la *Chanson de Roland*. L'article qui suit, fera partie du manuel de ce logiciel, mais ce n'est pas une simple description technique présentant comment utiliser celui-ci. C'est plutôt la présentation d'une méthode. Méthode pour décrire l'état du texte dans un manuscrit, avec le plus de simplicité et le plus d'exactitude. Mais notre méthode n'est pas encore, bien entendu, ni parfaite ni définitive. Elle doit être mise à l'examen aussi bien de chercheur qui utiliseront Scriptorium, que de ceux qui ne l'utiliseront pas et donc ne liront jamais son manuel. C'est pourquoi nous publions la première partie du manuel comme article.

ておく。

1 はじめに

本稿は、筆者が作成しつつあるテキスト校訂用ソフトウェア **Scriptorium** に付される、マニュアルの第一部である。まだ完成していないソフトウェアの、しかもマニュアルを論文とするのには異論もありえるだろう。しかし、本稿内で述べられるのは、単なるソフトウェアの使用法ではない。ソフトウェアは、処理対象を厳密に定義しなければならないが、目下の場合、そうした定義が校訂法と不可分なのである。実際、**Scriptorium** は、筆者の研究『『ロランの歌』データベースの構築と電子校訂法の確立』（科学研究費補助金、研究課題番号：21520331）の中で、生まれでたものであり、それ自体が、テキスト校訂の新しいあり方の提案、という側面を持っているのである。そうである以上、**Scriptorium** の仕様は、校訂法の観点からの批判が可能である。そして、そうした批判への途は、**Scriptorium** の利用者がいかに拘らず、あらゆる専門家に開かれねばならない。本稿を独立した論文として発表するゆえんである。なお、**Scriptorium** は作成途上であるが、一応の完成は見ており、すでに、筆者自身が使用を開始している。本論刊行までには、インターネット上にも公開できるよう研究日程も組んである (www.eonet.ne.jp/~ogurisu/)。

以下では、マニュアルへの転載を考慮して、です・ます調で書かれていることを、あらかじめお断りし

2 Scriptorium とは何か

以下では、テキストとテクストという二つの単語が交互に現れます。テキストは校訂対象となる作品の本文、写本内の読みといった、文献学な意味合いにおいてのみ使用されます。一方、テクストは、テクストファイル、電子テクストといった具合に、コンピュータ用語として用いられます。

2.0.1 概論

Scriptorium は作者が専門とする中世仏語の韻文作品、とりわけ武勲詩の校訂作業を効率的に進めるために開発されたソフトウェアです。工夫すれば散文や他ジャンルの作品校訂にも利用できるかも知れませんが、武勲詩以外のテキストは、今のところ、サポートの範囲外です。

Scriptorium は、コンピュータ上で動作するソフトウェアです。しかし、実際には、私たちが提案する校訂作業の方法論と表裏一体の関係にあります。したがって、本稿では、ソフトウェアの使い方以上に、原稿の準備の仕方や、**Scriptorium** の処理内容を説明します。他方、写本の読み方や校訂法の詳細については、必要に迫られない限り触れません。これらについては、下記の優れた書物を参照してください。

1. Alfred FOULET, Mary Blakely SPEER, *On Editing Old French texts*, The Re-

- gents Press of Kansas, 1979, Lawrence.
2. Yvan G. LEPAGE, *Guide de l'édition de textes en ancien français*, Honoré Champion, coll. "Moyen Âge - Outils de Synthèse", 2001, Paris.
3. Pascal BOURGAIN, FRANÇOIS VIELLIARD, *Conseils pour l'édition des textes médiévaux, fascicule III, textes littéraires*, Comité des travaux historiques et scientifiques, École nationale des chartes, 2002, Paris.

つまるところ、本稿で問題となるのは、写本をどう読むか、ということではなく、読み取ったテキストをどう記述するかということになります。もっと言えば、校訂者が、写本に基づいて、コンピュータに何を入力するか、ということです。コンピュータが校訂作業の強力なツールになることは、もはや常識

でしょう。しかし、コンピュータほど利用者の習熟度や方法論の違いで作業効率に差が出る道具はありません。ここで提案しようとするのは、一つの方法論に基づいた合理的なテキスト記述法です（単なるテキストの入力法ではないことにご注意ください）。

さて、その方法論のもととなる着想はきわめてシンプルです。コンピュータ上でのデータ操作に耐え、ディプロマティック版として出力でき、なおかつ、テキストクリティック版としても出力できるような、テキストファイルを校訂作業の時点で作成してしまう、というものです。単純化すれば、校訂作業の結果仕上がったテキストファイルを、アプリケーション上にドロップしさえすれば、ディプロマティック版やテキストクリティック版の透過原稿 pdf ファイルが作成できたり、コンコードランスや索引などを作成できるようにしよう、というのが私の提案です。

百聞は一見にしかず。次のテキストファイルをご覧ください。

元のテキストファイル

```
1ro 1  CARLES li reis, nostre em\per|[er]e magnes,
      Set anz tuz pleins ad estét en ESPAIGNE :
      Tresqu'en la mer \cun|quist la tere altaigne.
      N'i ad castel ki devant lui remaigne ;
      Mur ne citét n'i est remés a fraindre,
      Fors SARRAGUCE, ki est en une muntaigne.
      Li reis MARSILIE la tient ki DEU nen aimet,
      MAHUMET sert \e| APOLLIN recleimet :
      Ne's poet garder \que| mals ne l'i ateignet. AOI
```

何の変哲もないテキストファイルです。とはいえ、通常の校訂規則に反している部分もあります。固有名詞が大文字で綴られていたり、単語の一部が \ と | で挟まれていたりといった具合です。

このテキストファイルを、Scriptorium と $\text{\LaTeX}2_{\epsilon}$ で処理することにより、次のような出力結果を得ることができます（検索可能な透過原稿 pdf のファイルです）。

テキストクリティック版

```
[1ro]                                1
0001  CARLES li reis, nostre emper[er]e magnes,
0002  Set anz tuz pleins ad estét en ESPAIGNE :
0003  Tresqu'en la mer cunquist la tere altaigne.
0004  N'i ad castel ki devant lui remaigne ;
0005  Mur ne citét n'i est remés a fraindre,
0006  Fors SARRAGUCE, ki est en une muntaigne.
0007  Li reis MARSILIE la tient ki DEU nen aimet,
0008  MAHUMET sert e APOLLIN recleimet :
0009  Ne's poet garder que mals ne l'i ateignet. AOI
```

ディプロマティック版

[1r^o]

1

0001 carles li reis nostre empe magnes
 0002 set anz tuz pleins ad estet en espaigne
 0003 tresquen la mer équist la tere altaigne
 0004 ni ad castel ki deuant lui remaigne
 0005 mur ne citet ni est remes a fraindre
 0006 fors sarraguce ki est en une muntaigne
 0007 li reis marsilie la tient ki deu nen aimet
 0008 mahumet sert 7 apollin reclimet
 0009 ne s poet garder q̃ mals ne li ataignet aoi

テキストクリティック版は（以下、クリティック版）、通常の校訂本のような形式で出力されています。他方、ディプロマティック版は写本の状態をできる限り再現しようとしています。もう少し詳しく言いますと、**Scriptorium** が出力する二つの版は次のような特徴をもっています。

クリティック版では

1. 略号の解釈結果が斜体で表記されます。
2. 固有名詞がスモールキャピタルで表記されます

ディプロマティック版では

1. 大文字はすべて小文字に変換されます。
2. j は i に、v は u に変換されます。
3. アクサンや句読点類は全て取り除かれます。
4. 写本内で使用されている略号が再現されます。

ディプロマティック版は、基本的には、写本を再現するのですから、写字生に由来しない 1,2 の区別がなくなるのは当然のことです。とはいえ、実際の写本では大文字と小文字が混在していますから、文字を全て小文字に書き換えるのは、写本再現の原則とは相容れません。j-i や v-u の変換にも同じことが言えます。しかし、写字生による、これらの文字の使い分けは、ほとんどの場合、大きな意味を持ちません。つまり、関与的では有りません。したがって、これまでの校訂法の伝統に従い、ディプロマティック版では全てを小文字や i, u に変換するのが、一番手っ取り早いのです（ただし、後述の通り、ローマ数字は例外的な扱いを受けます）。逆から言えば、こうした表記の違いに着目した校訂を行うには、**Scriptorium** は全く適していません。ついでに付け加えておきましょう。単語の区切りについても中世の写字生と現代人とは基準が違いますが、単語の区切りに着目した校訂にも **Scriptorium** は対応していません。

さて、話を本筋に戻しましょう。目下、重要なのは、上記二つの版のいずれもが、先にあげたテキス

トファイル进行处理することで作成されるということです。したがって、テキストファイルを修正すれば、クリティック版もディプロマティック版も同時に修正することができるのです。校訂作業を行ったことのある人なら、そのメリットは十分に理解できるでしょう。

レイアウトされたクリティック版の修正は通常、大きな手間がかかります。それに、クリティック版出力用の原稿と、写本テキストを転写した原稿が別のファイルになっている場合、その両方を修正しなければなりません。Scriptorium を使えば、修正したテキストファイルから、整形されたクリティック版を作り直すことができます。とはいえ、もっと大きなメリットがあるのは、もちろん、ディプロマティック版です。これにより、写本と転写結果の比較が格段に容易になり、多くの校訂ミスをも未然に防ぐことができます。

また、原稿となるテキストファイルは、一貫した方法論にしていますから、データ処理が格段に容易です。クリティック版では、固有名詞がスモールキャピタルに、略号が斜体になっていましたが、テキストを見苦しくしていると思う人もあるかも知れません。実をいえば、これらの文字修飾を外してしまうのは、非常に簡単なことなのです。大事なのは、固有名詞や略号が、それとして見分けられている、という事実です。これにより、たとえば、**Scriptorium** は、略号の使用箇所を網羅的に示した索引や、固有名詞の索引を自動作成します。これらは校訂結果の見直しを進める際に欠かせないデータとなるでしょう。また、私が作成したもう一つのソフトウェア、LexicaNEO にそなわったコンコーダンス機能（HpConc 機能）は、Scriptorium 原稿の書式に対応しており、あらゆる単語の使用箇所をたちどころに表示することができます。これらのことが可能になるのは、もとのテキストファイルで、固有名詞が全て大文字で綴られ、略号が \ と | で挟まれていたからにほかなりません。先ほど、本稿で問題になる

のが、テキストの記述法であって、単なる入力法ではない、と述べたのは、まさしくこのことを言っているのです。

2.0.2 Scriptorium が要求する事柄

Scriptorium を利用するためには、次のことが必要です。

1. 電子テキストを作成した経験があること。
2. $\text{\LaTeX}2_{\epsilon}$ にある程度習熟していること。

第一の条件を説明しましょう。自分にとってだけでなく、他人にとっても使い勝手の良い電子テキストを作成するには、それなりのノウハウが必要です。もし、あなたがその経験を欠いているのであれば、電子テキストを一つ仕上げてください。拙論、『電子テキストの書式 ―より快適な電算処理のために―』が参考になるでしょう（ただし、この論文の文字コードに関する記述はもはや時代遅れです）。あなたがこれから校訂しようとしている作品の、過去の校訂本を電子化しておくのは、決して無駄な回り道ではないはずです。

第二の条件も欠かすことはできません。Scriptorium はあなたの原稿を処理して、複雑な構造をもった出力ファイルを作成します。それを書籍のような姿に整形するのは $\text{\LaTeX}2_{\epsilon}$ という組版ソフトウェア（フリーウェア）なのです。したがって、あなたのコンピュータには pdf 書類を出力できる $\text{\LaTeX}2_{\epsilon}$ 環境が構築されていなければなりませんし、あなた自身が $\text{\LaTeX}2_{\epsilon}$ を使用した経験も必要になります。もちろん、今からでも遅くありません。 $\text{\LaTeX}2_{\epsilon}$ をインストールし、あなたの過去の論文のいくつかを $\text{\LaTeX}2_{\epsilon}$ で組版し直してみれば良いでしょう。論文がすでにワープロなどの電子原稿になっていれば、組版をやり直すのは、それほど大変な作業ではありません。書店に行けば、 $\text{\LaTeX}2_{\epsilon}$ 関連の書籍は山のように見つかります。中でもおすすめなのは、奥村晴彦氏の『 $\text{\LaTeX}2_{\epsilon}$ 美文書作成入門』です（毎年のように改訂されますから、あえて書誌情報は記しません）。この本には $\text{\LaTeX}2_{\epsilon}$ をインストールするための CD-ROM が付属していますし、この本の最初の百ページほども読めば、論文を組版するくらいはできるようになります。最初のうちは何度も $\text{\LaTeX}2_{\epsilon}$ のエラーが出て、閉口するでしょうが、根気よく作業をしてください。慣れるにつれて、エラーは激減します。

以上の条件にうんざりした方もあるかも知れませんが、けれども、本当に大事な最後の条件が残っています。

- 3 中世古写本の校訂に関する知識を備えていること。

これは自明ではありますが、本マニュアルも Scriptorium も写本の読み方をあなたに教えることはありません

第三の条件を満たすことに比べれば、第一、第二の条件をクリアするのは、非常に容易いことです。どちらも一週間ほど集中して作業をすれば、会得できます（電子テキストの作業に要する時間は電子化するテキストの長さによって異なりますが、500 行ほどの作品なら一週間でなんとかなるでしょう）。そして、この二つの条件をクリアすることは、結局のところ、あなた自身のテキスト電算処理のスキルを高め、校訂作業だけではなく、研究や教育に関わる他の多くの局面で、あなたに大きな利益をもたらすことになります。たとえば、科研費の書類を作成する作業も、 $\text{\LaTeX}2_{\epsilon}$ を使ったほうがずっと時間の節約になるのです。

3 本文テキストファイルの作成

以下では、Scriptorium で処理するためのテキストファイルの作製法を紹介していきます。以下に述べられる、ルールに従うことで、校訂本出力や電算処理に適したやり方で、テキストを記述することができます。もちろん、校訂作業の最初期の段階から、全てのルールに則ったやりかたでテキストファイルを作成する必要はありません。どういう手順で、テキストファイルを作成するかは、その人の好みや方法論によるでしょう。たとえば、固有名詞を全て大文字で綴るのが手間な場合は、後から一括置換でうまく処理できる場合もあるでしょう。けれども、最終的に出来上がったテキストファイルが、以下のルールに則っていない場合、当然ながら、Scriptorium の処理で、期待した結果を得ることはできません。

3.1 書式

本文は、次の書式にしたがい、プレーンテキストファイルとして、入力します。文字コードには utf-8、改行は Unix のものを使用してください

1-フォリオ番号⇒2-詩節番号⇒3-詩行番号⇒4-詩行¶
(⇒はタブを¶は改行を表します)。

上記のようにタブで区切られた各部分を以下ではフィールドと呼びます。各フィールドは次のように入力します。

1. フォリオ番号は、フォリオの表裏やカラムが更新された行に、その参照番号や記号を二つの/(スラッシュ)の間にはさんで入れます。//だけを入力して、Scriptorium で処理する時

に自動的入力することもできます。

2. 詩節番号は新しい詩節の開始する詩行に、詩節番号を二つの/の間にはさんで記入します。//だけを記入しておいて、自動入力することもできます。
3. 詩行番号は、空白にしておけば自動入力されます。
4. 詩行：ここに原典の詩行を一行入力します。

次の点にご注意ください。

1. 一行に四つ以上のタブが入力されている時には、余分なフィールドは全て本文と見なされます（タブはスペース一個に置き換わります。つまり、レイアウトには使用できません）。
2. 詩節番号をローマ字で出力したり、詩行番号を一定のインターバルで出力したりといったことは、設定で対応できます。また、後述の通り、脱落行や重複行にも対処法があります。

3. 2で対応できないような事情がない限りは、フォリオ、詩節、詩行の番号は、自動入力を利用の方が確実に、レイアウトの自由度も向上します。

4. フォリオ番号、詩節番号、詩行番号のいずれも、番号を入力した場合でも、自動入力で書き換えることは可能です。

5. 番号を入力すべきフィールドでも、あなたの入力した文字列が尊重されます。番号以外のものが入力されていても、そのまま出力されます（当然ながら、自動入力で書き換えた場合には、その限りではありません）。

デフォルトでは、各フィールドは下記のようにレイアウトされます（テキストクリティック版の場合）。2r0 がフォリオ番号、0058...0062 が詩行番号、太字の 5 が詩節番号です。なお、番号の表記法は、Scriptorium の操作で、簡単に変更することができます。

[2r0]	0058	Asez est mielz qu'il i perdent les testes
	0059	Que nus perduns clere ESPAIGNE, la bele,
	0060	Ne nus aiuns les mals ne les souffraites »
	0061	Dient paien : « Issi poet il ben estre. »

5

0062	Li reis MARSILIE out sun cunseill finét,
------	--

3.2 詩行番号

Scriptorium の処理では詩行番号は非常に重要な意味を持ちます。本文を入力する時に、あなたが最も注意を払わなくてはならないのは、詩行番号をテキストファイルの段落番号と常に一致させることです。たとえば、写本のテキストを第 1501 詩行から校訂したい場合、テキストファイル冒頭の 1500 段落は空白のままにします。

なお、段落とは、改行文字もしくは文書終端で終わる文字列のことです。改行文字とは、**return** キーや **enter** キーを押すことで入力できる不可視文字で、段落の終端を表します。逆に段落の冒頭は、文書冒頭か、前の段落の終端の直後ということになります。

段落とワードラップをくれぐれも混同しないようにしてください。文字列を入力し続けると、カーソルは画面の右端に達し、左端に戻りますが、これはワードラップであり、段落の変更ではありません。**return** キーや **enter** キーを押さない限り、段落が改まることはありません。

原稿の入力には、改行文字や段落番号を表示できるようなテキストエディタを利用することを強くお

勧めします。

さて、作品によっては、詩行番号の数え方に伝統があり、写本の本当の詩行番号と校訂本で使用する詩行番号が必ずしも一致しません。伝統に従うのであれば、段落番号と一致させなくてはならないのは、校訂本で使用する詩行番号のほうです。この場合、次のことに注意してください。

1. 写本に対応する詩行がない場合は、原稿の該当段落を削除せず、空欄のままにします（...などを入れてもよいでしょう）。
2. 校訂本の詩行番号に、写本内の複数の詩行が対応する場合には、該当する段落の中に、すべての詩行をすべて詰め込みます（詩行の変わり目には//と二度打ちしたスラッシュを入れます。）。

前者の場合、詩行は空白のまま（あるいは... などが）出力され、詩行番号を自動カウントした場合にも、詩行としてカウントされます。

後者の場合、すべての詩行がアルファベット文字付きの行番号となります。たとえば、次のようになります。

元のテキストファイル

```
0001 CARLES li reis, nostre em\per[er]e magnes, ¶
0002 Set anz tuz pleins ad estét en ESPAIGNE ://Tresqu'en la mer
\cun|quist la tere altaigne.//N'i ad castel ki devant lui remaigne ; ¶
0003 Mur ne citét n'i est remés a fraindre, ¶
```

テキストクリティック版

```
0001 CARLES li reis, nostre emper[er]e magnes,
0002 Set anz tuz pleins ad estét en ESPAIGNE :
      a Tresqu'en la mer cunquist la tere altaigne.
      b N'i ad castel ki devant lui remaigne ;
0003 Mur ne citét n'i est remés a fraindre,
```

3.3 詩行の入力

詩行は通常の電子テキストと同様の方法で入力できます。それゆえ、あなたの手元にある電子テキストを、ここまで述べた書式に則って整形すれば、そのまま **Scriptorium** と **L^AT_EX2_ε** で処理し、テキストクリティック版を作成することができます (うまく行かない場合には、本稿を読み終えてから、再挑戦してください)。ただし、本稿冒頭に示したような出力結果が得られるわけではありません。斜体やスモールキャピタルになっている部分は、一つもないはず。これらは、それぞれ固有名詞と略号の解釈結果を強調表示するために用いられます。しかし、通常の電子テキストには、そうした部分を見分けるための情報が何一つ書き込まれていませんから、**Scriptorium** は、一つの固有名詞も、一つの略号解釈も、存在しないと見なして、処理を進めたのです。したがって、**Scriptorium** の能力を十全に利用するためには、電子テキストを、いくつかのルールにしたがって、書き換える必要があります。以下では、そのルールを説明します。

3.4 本文内で使用できない文字

下記の文字を除いて、キーボードから直接入力できる文字のすべてが利用できます (これらは **L^AT_EX2_ε** のコマンド、もしくは、**Scriptorium** の特殊な用途に利用されます。詳しくは 7.7 の「使用可能文字」をご覧ください)。

\$ % & _ { } \ ^ ~ |

また、次項に記す通り、(), [] の使用法にも制限があります。最後に重要なことですが、本文ファイル内では、**L^AT_EX2_ε** コマンドは、使用できません。

3.5 テキストの追加と削除

写本のテキスト内の単語を削除したり、追加したりする場合には、括弧を使います。削除すべき単語は () で、追加すべき単語は [] で挟みます。これらの単語は、クリティック版では括弧がついたままの状態です。それにより、読者は校訂者が写本のテキストを変更した箇所を簡単に見分けることができます。一方、ディプロマティック版では、写本の内容を再現するために、() 内の単語は括弧がはずれた状態で出力されます。そして、[] 内の単語は括弧共々削除されます。

ただし、上記の方式では、読めない単語 (文字) を補った場合に対応できません。削られて (*gratage*)、あるいは汚れにより読めなくなった単語 (文字) を、単に削除したのでは、写本の状態が再現されたことにはなりません。

一例を挙げましょう。Gérard Moignet の *La Chanson de Roland* (Bordas, 1989, Paris) の第 3595 行には、« **sembl** » という奇妙な形の動詞が見られます。実は、これは二種類の写本を合本して、サイズを揃えるために、小口が裁ち落とされた際に、単語の末尾が失われたもののなのです。これは、誰の目にも明らかな事実です。しかし、Moignet は、« **sembl** » をそのままにしたばかりか、ご丁寧に、その後にポワン・ヴィルギュルを付しています。これでは、読者は写字生が « **sembl** » という単語を実際に書いたと思い込んでしまうでしょう。

さて、この « **sembl** » を **sembl[et]** と入力した場合、ディプロマティック版では、**sembl** と表記されてしまい、Moignet 氏の轍を踏むことになってしまいます。それを避けるために用意したのが、[[]] です。

[[]] の中の単語はクリティック版では【と】で挟まれます。ディプロマティック版では、[[]] とともに、□□□ になります。なお、□ は [[と]] に挟まれた文字

の数だけ出力されます。この方式はディプロマティック版を見栄え良くします。li che[**(valiers)**] est が li che┐ est となるよりも、li che┐┐┐┐┐┐┐┐┐┐ est となる方が、おそらくは、写本の状態にも近いことでしょう。けれども、この方式は、脱落した文字の数に関して、読者に先入観を持たせる恐れもあります。補った文字の数が、写本の不明部分の大きさと一致しない場合には注釈でそのことを断るべきでしょう。

入力	[chanson]	(de)	[(roland)]
C 版	[chanson]	(de)	[roland]
D 版		de	┐┐┐┐┐┐┐┐

同じやり方で、文字を補うこともできます。

入力	cha[n]son	de(e)	ro[(la)]nd
C 版	cha[n]son	de(e)	ro [la]nd
D 版	chason	de	ro┐┐┐nd

なお、これまでに述べた意味で括弧類を使用する場合には、複数の単語は、その一つ一つを括弧でくくってください。

入力	(chanson) (de) (roland)
C 版	(chanson) (de) (roland)
D 版	chanson de roland

これは、() と [] を通常の使い方でも使えるようにするためです。特に、() は中世の作家な自由な構文法を理解可能なものにするために、多くの校訂者が利用しています（その場合、単語を一つだけ括弧に括るということは通常ありません）。単語や文字の追加・削除以外の意味で、[] や () を利用するためには、(())、[[]] という風に二度書きします。これらは、クリティック版では (), [] になります。一方、ディプロマティック版では削除されます。

3.6 固有名詞

全て大文字で綴られた二文字以上の単語（ローマ数字を除く）を、Scriptorium は固有名詞として認識します。固有名詞はクリティック版ではスモールキャピタルで組まれ、ディプロマティック版では下線が引かれます。また、固有名詞索引に登録されます。[, (, \, | は大文字の扱いとなり、RO[LA]ND, RO[(LA)]ND, ROLAND(D), \ROLAND| のいずれも固有名詞として扱われます。

3.7 ローマ数字

ローマ数字そのものは、大文字のみで表記します。次に、マーキングですが、ローマ数字は終止符で挟む

習慣です。が、これは決して望ましい習慣ではありません（終止符は文の終わりだけに用いるべきですし、写本内の点の位置は、終止符のそれとは明らかに異なります）。・CC・IIII・というふうにセンタードットで挟むようにします。センタードットはアスキー番号 Mac-225, Win-183 の文字です。Macintosh の場合、「オプション+シフト+9」で入力できます。なお、センタードットを、Mac-165, Win-149 の大きなドットと混同しないようにご注意ください。入力に自信がない場合には、#や～などで代用しておいて、最後に一括置換すると良いでしょう。正しい記号は、Scriptorium の画面上で、コピーすることができます。

以上のようにローマ数字をマーキングすることで、クリティック版では、ローマ数字がスモールキャピタルで出力され、ディプロマティック版では、V が U に書き換えられるのを防ぐことができます。なお、・VII・C・ と ・VIIC・ はどちらも 700 ですが、校訂本上では、・VII・C・、・VII^C・ と表記が異なってきます。・IV・XX・ は、・IV^{XX}・ としないと、混乱を招くかも知れません。

3.8 前接語

前接語は、古いフランス語の初心者にとっても厄介ですが、コンピュータにとっても困った存在です。Atkinson Jenkins や Cesare Segre は *Chanson de Roland* の刊行本において、sis、sil といった連語を si・s、si・l などと表記しています。この方式は、上記の問題に対する一つの解決法と言えるでしょう。Scriptorium は、si●s、si●l という入力を si・s、si・l と組版します。とはいえ、al、als、del、dels を a・l、a・ls、de・l、de・ls とするのは、電算処理の観点からは望ましいかも知れませんが、あまりに伝統に反しており、お勧めしにくいところです。そこで、Scriptorium は●を無視して組版することもできるようになっています。その場合、si●s、si●l、a●ls、de●ls... は、sis、sil、als、dels と組版されます。

なお、●[ビュリット、(Mac-165, Win-149)] をローマ数字の桁を区切った・[=センタードット (Mac-225, Win-183)] と混同しないようにしてください。●は Macintosh の場合、「オプション+8」で入力できます。入力に自信がない場合には、#や～などで代用しておいて、最後に一括置換すると良いでしょう。正しい記号は、Scriptorium の画面上で、コピーすることができます。

3.9 略号 (概要)

写本内では通常多数の略号が使われます。これらの略号を表すには、略号の読みを \ と | ではさみます。たとえば、**p** に **per** という読みを対応させて、`\per|` と入力すれば、クリティック版では **per** が斜体に、ディプロマティック版では略号が出力されます。

入力	<code>em\per ere</code>
C 版	<code>emperere</code>
D 版	<code>empere</code>

しかし、これはまだ概略に過ぎません。実際に略号の表記と表示を行うにはいくつかの手続きが必要です。それについては次節で検討しましょう。

4 略号

4.1 最初に

前節では、略号の読みを \ と | で挟むことで、写本内の略号を扱うことができるということを述べました。 \ と | 挟まれた文字列は、**Scriptorium** にとっては一種の命令であり、その命令にしたがって、**Scriptorium** は、文字列を斜体で出力したり (クリティック版)、略号そのものを出力したり (ディプロマティック版) します。

しかし、**Scriptorium** のこの機能を利用するためには二つの課題をクリアしなければなりません。

1. 略号とその読みを適切に対応させる。
2. 略号を出力するための仕組み (マクロ) を作成する。

一見、大変そうなこれらの作業は、多少時間を要するにしても、正しい手続きにのっとれば、それほど難しいものではありません。そして、作業に要した手間に見合うだけの、メリットを享受することもできます。写本のテキストと見比べてミスを見つけようとする際、ディプロマティック版の方が、クリティック版よりもずっと効率良く作業できますし、ミスの発見も容易なのです。

4.2 略号の読み

略号の読み方を \ と | で挟むことで、略号を表す命令とするというのは、基本的には誰もが納得できるシステムでしょう。たとえば、**q** を `\qui|` と表記することに大きな違和感を覚える人はいないでしょう。しかし、いくつかの疑問を感じる方もいらっしゃるはずです。それは次のようなものでしょう。

1. 全ての写本で略号の読み方が一定しているわけではない。
2. 一つの写本の中でも、複数の読み方をする略号が存在する。
3. 一つの写本の中で、同じ読み方をする略号が、複数存在する。

第一の問題点により、略号とその読みの一定した対応表を作成することが不可能になります。したがって、略号表記システムを完全に既製品化することは不可能になりますが、これについては、後で略号表記マクロを扱う際に詳述することにしましょう。目下重要なのは、写本ごとに略号とその読みの対応表を、あなた自身が作らなければならない、ということです。しかし、これは問題とは言えません。これは写本を読めば必ず行わねばならない作業です。むしろ、私たちの提案するシステムを採用することで、この作業をより正確に行えるようになるということだけを、今は指摘しておきましょう。

第二の問題点、一つの写本の中でも、複数の読み方をする略号が存在する、も大きな問題とはなりません。一つの略号に複数の名前を付ければ良いのです。たとえば、**p** に `\per|` と `\par|` という二つの名前を付けても全く問題ありません (三つ以上でも構いません)。

逆に第三の問題点は、エレガントな解決が困難です。たとえば、**que** を表す、二つの略号 `\q̄|` と `\q̇|` の両方を、`\quel` で表してしまうと、**Scriptorium** は、ディプロマティック版を作成する際、`\quel` を `\q̄|` と `\q̇|` のどちらで置き換えてよいのかが区別できなくなります。したがって、番号付きの名前を利用するほかありません。`\q̄|` を `\que1|`、`\q̇|` を `\que2|` といった具合です。しかし、これは、もはや略号の命名規則ですから、次項で説明することにしましょう。

4.3 略号の命名規則

既に述べた通り、 \ と | に挟まれた文字列は、クリティック版ではそのまま斜体で出力されます。たとえば、`emperere` といった具合です。これにより、読者は **per** が写本の中では略号で表されているのだと知ることができます。したがって、略号の命名をする時の第一のポイントは、クリティック版に表記して良いような名前をつける、ということです。**p** を `\pbar|` などと名付けても **Scriptorium** は何の文句も言いませんが、`empbarere` では、クリティック版の名に値しません。

なお、すでに述べたことですが、ディプロマティック版では、

1. 大文字はすべて小文字に変換されます。

2. j は i に v は u に変換されます。
3. アクサンや句読点類は全て取り除かれます。

したがって、`\par` と `\Par` の間、`\que` と `\qué` の間、`\ver` と `\uer` の間には、違いがありません。ディプロマティック版では、これらの命令で同じ略号が出力されることになります。

以上は Scriptorium の仕様に基づいた記述であり、厳密な意味での規則です。これを無視して略号命名を行っても、意図した出力結果を得ることは決してできません。一方、以下に述べるのは、いわばガイドラインです。「命名規則」というよりも「命名法」と呼ぶべきでしょう。それぞれの写本の事情もあるでしょうから、私たちの提案には、必ずしも従う必要はありません。しかし、従わない場合には、略号出力用の $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ マクロを自前で準備しなければなくなることもあります（そうでない場合もあります）。

私たちが提案する命名法の基本は、略号と文字を切り離さない、というものです。たとえば、**p** の横棒だけを取り出して、**er** や **ar** の略号だとは考えずに、**p** と横棒の組み合わせ全体で、**par** や **per** の略号だと考えます。この命名法の妥当性は、**q̇** を例に考えると、納得しやすいでしょう。**q** の上の **i** が **ui** の略号だと感じる人は少ないはずです。**ṫ** が **tui** ではなく、通常は **tri** と読まれることなどを考えれば、なおさらです。

とはいえ、例外もあります。**ẽ** や **ã** を `\en|`、`\an|` とするのは、上の規則との整合性を考えるなら、きわめて合理的です。しかし、鼻子音を表す符号は頻繁に用いられますので、クリティック版を作成する際に、斜体文字になる部分が爆発的に増えます。これはテキストの見栄えを悪くし、視認性を低下させる恐れがあります。また、余分に略号命令を定義しなければならなくなります。それゆえ、次の結論となります。

鼻子音を表す記号は、**m**、**n** を表す符号と見なす。

つまり、**ẽ** や **ã** を `e\ñ|`、`a\ñ|` とします。ただし、上にあげたデメリットを甘受する限りにおいて、この命名法に従わなくても、つまり、`\en|`、`\an|` を採用しても全く問題ありません。

4.4 略号の命名手順

略号の読みを決定するには、略号の使用傾向や、略号が使用された単語に対応するフルスペルの単語との比較対照を行う必要があります。フルスペルの単語が複数の綴りを使用している場合、どちらの綴りの頻度が高いかも調べなければなりません。つまり、略号をどう読むかを決定するのは、本来、写本を転写

し終えた後です。しかし、それは「最終決定」の話です。とりあえずは、暫定的な命名を行うのが実際的です。そして、できあがった原稿を、Scriptorium で処理すれば、様々なファイルが出力されます。これらのファイルを利用したり（中でも、後述する略号使用箇所の網羅的索引は威力を発揮するはずですが）、テキストエディタの置換機能を使って、後から統一をはかるのです。その方が結局のところ、ミスが減りますし、手間も少なくて済みます。暫定的な命名については、具体的には、次の点に注意します。

1. どの読み方が正しいかで最初から悩まない：正しい読みは転写作業が終わってからしか決定できませんから、**&** を **e** と読むか、**et** と読むかで悩むよりも、とりあえず、**et** と読むことにして作業を進めた方が得策です。
2. 略号の読み方の一貫性に過度にこだわらない：前置詞の **p** を `\per|` と `\par|` の二通りの方法で表記していることに気づいたからといって慌てて統一する必要はありません。
3. 略号に最初から番号をつけない：同じ読み方をする略号が複数ある場合にも、最初はその名前前で表記しておきます。**q̇** も **q̈** も `\que|` としておいて、番号をつけるのは後回しにします。

もちろん、気づいたことはメモしておき、後で作業に役立てるべきなのは言うまでもありません。テキスト校訂は時間のかかる作業なので、何を後回しにしたのかを書き留めておかないと、作業を後回しにしたことを忘れてしまいかねません。

5 テクストクリティック版の処理

原稿の入力が終わったら、いよいよ Scriptorium と $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ による処理を行います。残念ながら、この処理が何の問題もなしに済むことはまずありません。特に慣れない間は、相当ストレスフルな作業を強いられることを覚悟するべきです。

5.1 本文ファイルの処理

5.2 Scriptorium の設定

Scriptorium を起動したら、まず、画面上の設定を済ませてしまいましょう。

現時点では、脚注や後注の書類は指定しないでおきます。設定は後から書き換えることができますので、とりあえず、処理対象の書類として本文用ファイルだけを指定します。次に、保存先のフォルダも指定してください。

最後に、番号の付け方や、注釈の形式などを設定し終えたら、設定を保存しておきましょう。設定は処理対象の書類が入っているフォルダに保存しておくとうまいでしょう。今後、**Scriptorium** を起動すれば、自動的に設定書類が読み込まれます（ただし、設定書類を別の場所に移動すると、自動読み込みはされなくなります。また、新たに、別の設定を他の場所に保存すれば、次の起動からは新しい設定の方が読み込まれます）。

5.3 Scriptorium の実行

Compiler ボタンを押せば、**Scriptorium** は処理を開始し、処理が終わったら、出力書類の入ったフォルダを自動的に開きます。そこには、**temp-files** と **result-files** という二つのフォルダが作られ、書類が分類して納められています。今から、あなたが $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ で処理すべき書類は **result-files** に、入っています。

5.4 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ による処理

5.4.1 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ の実行

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ による処理とは、指定された書類を対象に、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ を実行し、画面表示や印刷が可能な、pdf 書類を生成することを意味します。タイプセット、コンパイル、組版などとも言います。以下の作業では、この処理を頻繁に行いますから、処理を簡単に行える補助ソフトウェアを利用すると良いでしょう。**TeXWorks** は様々な OS 上で、**TeXShop** は **MacOSX** 上で、動作します。また、処理をおこなうために必要なパッケージが、あなたの $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ に組み込まれていない場合があります。たとえば、**LaTeX Error : File 'ledmac.sty' not found.** という表示が出たら、あなたの $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ に **ledmac** パッケージを組み込む必要があります。パッケージの追加方法に付いては、マニュアル本などを参考にしてください。なお、**MacOSX** ユーザーに関しては、www.tug.org/mactex/ でダウンロードできる **The MacTeX-2010 Distribution** には、日本語は扱うことができないという欠点があるものの、必要なパッケージが全て組み込まれているということを付け加えておきましょう（日本語用との共存は難しいので、私自身は、二台のコンピュータにわけてインストールしています）。

5.4.2 **PrintFile.critique.tex** の処理

まず、処理するのは、**PrintFile.critique.tex** という書類です。書類名のドット以下の部分は拡張子と呼ばれるもので、あなたのコンピュータの設定によっ

ては表示されないこともあります。以下、どの書類名に関しても同じことが当てはまります。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ のような **Unix** 系のソフトウェアを使用する場合、OS の設定を変更して、拡張子を表示するようにした方が圧倒的に便利です（ただし、拡張子の概念を十分に理解しておく必要があります）。

処理を始めたら、多くの場合、あっという間に、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ が処理を中断して、書類の不具合を訴えます。「書類の行目の... が理解不能である」といったような指示が出ますので、どこに不具合があったのかを、**PrintFile.critique.tex** 内で突き止めます（**TeXShop** では「エラー箇所」というボタンを押せば、問題の箇所にジャンプしてくれます）。次に、行番号をたよりに、本文テキストファイル（あなたが作成したテキストファイル）の該当箇所を探し出します。たいていの場合、じっくりと眺めれば、どこに不具合があったかは、すぐにわかります。次のような可能性を真っ先に疑ってみると良いでしょう。

1. 括弧類の照応が悪い。開き括弧もしくは閉じ括弧が脱落している。
2. 略号の \ や | が抜けている。
3. \$ や ~ などの使ってはいけない記号を使っている（3.4 「本文内で使用できない文字」）。

同じ行に複数のミスが生じている場合もあるので、その点にも注意してください。

さて、本文テキストファイルを正しく修正したら、再び **PrintFile.critique.tex** に戻ります。ミスの修正は本文テキストファイルの方で行ったので、**PrintFile.critique.tex** には、ミスが残ったままです。そのミスがある箇所の行頭に % を書き込んでください。これはコメントアウトとよばれるもので、% 以後から改行までを $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ は「コメント」と見なし、処理しないようになります。結果的に、不具合のある箇所を跳ばして、処理を実行することができるようになります（ミスが単純なものでない場合には、コメントアウトを使わずに、「修正-Scriptorium の処理- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ 」の処理を繰り返して、試行錯誤した方が良いでしょう）。以上の手続きを終えたら、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ での、**PrintFile.critique.tex** の処理を再び行ってください。

このように、ミスを修正しては、処理を再実行するというサイクルを、何十回、何百回、何千回と繰り返します。その回数は、コンピューターや $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ に対するあなたの習熟度により様々です（習熟すれば、確実にエラーの回数は減っていきます）。そして、ついには、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ が処理を完遂する時がきます。とはいえ、勘違いしてはなりません。あなたが修正したのは、本文ファイルの方ですから、**PrintFile.critique.tex** の方の問題は根本的には解決されて

いません（コメントアウトによって、不具合のある箇所をバイパス回避しただけです）。次に何をやるべきかはもはや明らかでしょう。Scriptorium の処理を再度行い、「正しい」PrintFile_critique.tex を作り直すのです。そして、もう一度、 $\text{\LaTeX}2_{\epsilon}$ の処理をやり直します。場合によっては、まだエラーが見つかるかもしれません（同一行内に二つのエラーがあった場合などに、二つ目のエラーを見落としてしまうこともあるからです）。その場合、ここまでの手順をもう一度繰り返してください。

最終的には、コメントアウトが全くない PrintFile_critique.tex を、 $\text{\LaTeX}2_{\epsilon}$ が滞りなく処理し終える時がやってきます。ここで、もう一度、 $\text{\LaTeX}2_{\epsilon}$ による処理を行ってください（Scriptorium が生成した書類を、完璧に組版するためには、 $\text{\LaTeX}2_{\epsilon}$ 処理を二回繰り返す必要があることを覚えておいてください）。

できあがった、PrintFile_critique.pdf を開いてみましょう。各ページを見て、期待した通りの組版結果になっているかどうかを確認してください。

エラー表示がないからといって、全てが期待通りというわけには、必ずしもいきません。というのも、 $\text{\LaTeX}2_{\epsilon}$ がエラーを表示するのは、原則として、与えられた命令が矛盾したり規則に反した場合だけです。人間の目にはどれほど不自然な組版結果になろうとも、命令に矛盾がなければ、 $\text{\LaTeX}2_{\epsilon}$ は黙って命令を実行します。たとえば、タブが一つ抜けたばかりに、詩行番号の入るべきフィールドに、詩行そのものが入ってしまっても、 $\text{\LaTeX}2_{\epsilon}$ は何のエラーも表示せずに、組版してしまいます。ですから、あなた自身の目で、組版の仕上がりを確認しなければならないのです。そして、おかしい部分があれば、本文テキストファイルに何らかの間違いがあるはずです。行番号をたよりに、本文テキストファイルのミスを修正してください。

上記に述べた処理手順は、今後の $\text{\LaTeX}2_{\epsilon}$ 処理すべてに通じる基本原則ですので、しっかりと身につけて覚えてください。

5.4.3 TablesDesAbbreviations.tex の処理

TablesDesAbbreviations.tex を $\text{\LaTeX}2_{\epsilon}$ で処理してみてください（処理が二回必要なことをお忘れなく）。今度は驚くほどスムーズに処理が終了するはずです。できあがった pdf 書類には、あなたが使用した略号がすべて収録されています。そして、それぞれの略号に対し、参照用の行番号が記載されています。TablesDesAbbreviations は、網羅的な「略号索引」なのです。

ただし、この「略号索引」は、未完成です。肝心

の略号がどこにも記載されていません。それもそのはず、略号を表示するための命令をまだ作成していませんから、略号が入るべき部分はすべて INDEF となっているのです。とはいえ、現時点でも略号の読み方に基づいた略号名が記載されていますから、索引は十分に使い物になります。参照行番号をよりどころにすれば、同じ読みをする複数の略号に、番号を付したり、間違った略号を修正したりといった作業が格段に楽になります。

ところで、TablesDesAbbreviations.tex を修正しても、略号を修正したことにはならない、ということは、強調しておかねばなりません。

例を挙げて考えてみましょう。索引に `\peer|` という「奇妙な」略号が見つかったとします。`\per|` と入力すべきところを、`\peer|` と綴ってしまったミススペルです。けれども、ここで、TablesDesAbbreviations.tex の `\peer|` を書き換えるのは無意味です。PrintFile_critique.tex を修正するのも同様です。修正すべきは、あなたが作成している本文テキストファイル（あなたが入力したテキストファイル）です。索引の参照行番号をたよりに、`\peer|` を見つけ出して、`\per|` と修正し、Scriptorium で処理をやりなせば、PrintFile_critique.tex も TablesDesAbbreviations.tex も正しく書きかわります。さらに、 $\text{\LaTeX}2_{\epsilon}$ で処理をやり直せば、索引からは、あの奇妙な `\peer|` が姿を消します。

番号付きの略号に関しても同じことがいえます。たとえば、`com` が `g` と `c` の二通りの略号で表記されている場合を考えてみましょう。この場合も、TablesDesAbbreviations.tex や PrintFile_critique.tex を書き直すのではなく、本文テキストファイルを修正します。手順としては、まず、写本画像で確認しながら、`c` の場合だけ索引の参照行番号に印をつけていきます。次に、その印を手がかりにして、本文テキストファイルの `\com|` に適切な番号を付けます。写本画像に行番号が付されていれば、この作業はずいぶんと楽なものになりますが、そうでない場合でも、temp-files 内に入っている行番号とフォリオ番号の対照表 (folio-vers.txt) を使えば、比較的容易に作業を進めることができるでしょう（行番号を入力すれば、該当する写本画像を表示するソフトウェアも開発中です）。

このように、略号を INDEF の状態のままにしても、略号の番号付けや略号のスペルミスの修正を行うことはできます。けれども、それ以上の作業、たとえば、略号の見落としや、不適切な略号解釈、略号解釈の不整合などの割り出し・検討については、略号表記のための命令（略号マクロ）を定義し、ディプロマティック版を作成したうえで、行った方が良いでしょう。写本との照合は、ディプロマティック

版を用いた方が容易だからです。次項では略号命令の作製法をみていきましょう。

5.5 略号マクロ

5.5.1 マクロとは何か

略号を出力するためには、 \LaTeX 2_ϵ のマクロの機能を利用します。

マクロというのは、簡単にいえば、ある命令文を名前と呼び出して実行する仕組みのことです。たとえば、ワープロでの作業を思い浮かべてみましょう。節の題名をイタリックの太字にして、14 ポイントの大きさにする、とします。これを、いちいちマウス操作で行うのは厄介です。また、後から気が変わって、文字サイズを 16 ポイントにしようと思ったら、またマウス操作をやりなおさなければなりません。一方、節の題名に、**SETSU** というマーキングをしておいて、後から、そのマーキングのついた部分を、「イタリックの太字にして、14 ポイントの大きさにする」と命令すれば、手間はずいぶんと軽減されます。文字サイズを変更したくなった場合も、命令文を書き換えれば事足ります。これがマクロと呼ばれる仕組みです。目下のたとえば、**SETSU** はマクロの名前で、「イタリックにして、太字にして、14 ポイントの大きさにする」はマクロの定義です。

本文ファイルを作る際、略号は、その読み方を略号名とし、`\` と `|` で挟む、というのが決まりでした。実は、この略号名はマクロ名としても機能します。ディプロマティック版を組版する時には、`\` と `|` で挟まれた「名前」に対応した命令を実行して、略号を描き出すのです。ただし、それを実現するためには、略号を描き出すための命令を別途準備しなければなりません。本節で解説するのは、そうした命令の定義の仕方です。

5.5.2 Abbreviations.tex 書類

先ほど作成した略語索引では、肝心の略語の部分は **INDEF** となっていました。これは、略号を描き出すための命令が未定義だということを表していました。しかし、**INDEF** という文字列を出力するという定義はなされているはずです。その定義が納められているのが、**Abbreviations.tex** 書類です。それを開いてみると、次のようになっています。

```
\def\DIPABBber{INDEF}%\ber|
\def\DIPABBcri{INDEF}%\cri|
\def\DIPABBcum{INDEF}%\cum|
...
```

もちろん、あなたの **Abbreviations.tex** が上の図と厳密に同じであるわけではありません。本文でど

んな略号が使われるかで、細かい部分は変わってきます。大事なものは、次の点です。

1. % の後に来るのが、あなたの使った略号名です。
2. **INDEF** の部分に、% の後の略号名に対応する略語を出力する定義を入れます。

ために、一番上の略号 **INDEF** を `\supra{q}{a}` とし、先の略号索引書類 (**TablesDesAbbreviations.tex**) を \LaTeX 2_ϵ で処理してみてください。出来上がった略号索引の最初の略号が、^aq となっているはずです。

次の二点を補足しておきましょう。

1. \LaTeX 2_ϵ で処理するのが、手を加えた **Abbreviations.tex** ではなく、**TablesDesAbbreviations.tex** だということに注意してください。前者は命令の定義が入っているだけです。
2. \LaTeX 2_ϵ がエラーになる場合には、入力した命令の綴りにミスはないか、**INDEF** を消す時に、前後の括弧を消してしまっていないかを確認してください。

以上が、略号マクロ定義の手順です。慣れないうちは、一つの定義を書き換えるごとに **TablesDesAbbreviations.tex** を \LaTeX 2_ϵ で処理しなおすと良いでしょう。 \LaTeX 2_ϵ がエラーになれば、今入力したばかりの定義に間違いがあることになります。混乱してきて、もとの **INDEF** の状態にまで、戻りたい時には、書類の末尾に、% でコメントアウトされた、バックアップがついています。ここから、コピーして、仕切り直したい行に上書きすると良いでしょう (行頭の % を外すのを忘れないでください)。さらに、**TablesDesAbbreviations.tex** を完全に初期化したい場合には、それをゴミ箱に捨て、**Scriptorium** での処理を再実行してください。新しい **TablesDesAbbreviations.tex** が自動的に生成されます。以上で、略号マクロの定義手順の説明は終わりです。具体的な定義の書き方は、次項で説明します。

5.5.3 マクロの定義

マクロの定義には、 \LaTeX 2_ϵ の命令を使用します。マクロ定義は、一から書くとなるとかなり大変な場合も少なくありません。そこで、**Scriptorium** はあらかじめ、マクロを定義するための下記のマクロ (上位マクロ) を準備しています。

<code>\supra{q}{e}</code>	$\overset{e}{q}$	上にのせる
<code>\infra{a}{c}</code>	$\underset{c}{a}$	下につける
<code>\trans{p}{---}</code>	\underline{p}	貫く
<code>\dextra{p}{o}</code>	$\underset{o}{p}$	右下
<code>\sinistra{p}{o}</code>	$\underset{o}{p}$	左下
<code>\Dextra{p}{o}</code>	$\overset{o}{p}$	右上
<code>\Sinistra{p}{o}</code>	$\overset{o}{p}$	左上

これらの上位マクロを用いることで、比較的簡単に二つの文字や文字と記号を組み合わせた略号が作成できます。やり方はとても簡単です。略号定義 (Abbreviations.tex) の INDEF の代わりに、表の第一列目の、いずれかの行の命令をそのまま書き込みます。その際、`{ }` と `}` の間の文字を書き換えれば良いのです。たとえば、`\supra{q}{e}` を `\supra{q}{a}` と書き換えれば、 $\overset{e}{q}$ が $\overset{a}{q}$ となります。

上の上位マクロのうち一番頻度の高い `supra` 他には下記のサブセットもあります。

<code>\asupra{q}</code>	$\overset{a}{q}$	a をのせる
<code>\esupra{q}</code>	$\overset{e}{q}$	e をのせる
<code>\isupra{q}</code>	$\overset{i}{q}$	i をのせる
<code>\usupra{q}</code>	$\overset{u}{q}$	u をのせる
<code>\osupra{q}</code>	$\overset{o}{q}$	o をのせる
<code>\rsupra{q}</code>	$\overset{r}{q}$	r をのせる
<code>\omegasupra{q}</code>	$\overset{\omega}{q}$	ω をのせる
<code>\lineatrans{s}</code>	$\text{\textit{s}}$	一で貫く
<code>\titulustrans{l}</code>	$\text{\textit{l}}$	～で貫く

下記は別系列の上位マクロで、複数の文字列に対応した長さの変わる、ラインとティルドです。どちらも一つの文字に利用した場合、通常の長音記号やティルドのついた文字とは若干おもむきが異なります (表内第二列と第三列を比較)。

<code>\lineasupra{abc}</code>	\overline{abc}	
<code>\lineasupra{a}</code>	\overline{a}	cf. <code>\=a \tilde{a}</code>
<code>\titulussupra{abc}</code>	\overline{abc}	
<code>\titulussupra{a}</code>	\overline{a}	cf. <code>\~a \tilde{a}</code>

下記も別系列の上位マクロで、使用頻度が高いけれども、作成するのが難しいと思われる記号です (見ての通り、今までの場合と違って、表の第一列の全ての部分が命令ではありません)。

R. <code>\bassiarenum{7} 0.</code>	R. 7 O.	文字を下げる
<code>a\titulusante</code>	\tilde{a}	直前の文字にティルド
<code>u\buculamini</code>	$\underset{u}{9}$	直前の文字に ⁹
<code>\buculamaj</code>	9	

一つ目は主に数字を字下げして記号に見せます。上記は Oxford 本 *Chanson de Roland* などにみられる接続詞 `e` の略号ですが、これ以外にも `de` が数字の 8 のように見えることもあります。Chanson g Roland とすると、略号に見えます。二つ目は、鼻母音の `\m`, `\n` に利用します。三つ目と四つ目に関しては、説明の必要はないでしょう。

ところで、必要なマクロが上記だけでそろわない場合にはどうすれば良いのでしょうか。L^AT_EX₂_ε 標準あるいは、amsmath パッケージが提供する数式用のマクロはかなり柔軟ですし、習得も難しくありません。簡単なものならば、それらを利用して、自前のマクロ定義を作成することができます。困難に出くわしたら、筆者にコンタクトをとるのも一つの方法です。可能な限りの協力を惜しまないでしょう。

その一方、私が提供するマクロが洗練を欠いたものであることもまた認めておかねばなりません。より柔軟で美しい出力を実現するマクロの提供を、筆者が待ち望んでいるということも付け加えておきましょう。

5.5.4 TablesDesNomsPropres.tex の処理

略号マクロの定義が完全に終わり、略号索引が完全な形で出力できるようになったら、次に、TablesDesNomsPropres.tex を L^AT_EX₂_ε で処理してみましょう。ここまでの作業を完璧にこなしていれば、処理はエラーなしに終わります。なお、固有名詞索引は、綴り字ミスや略号の解釈ミスを発見するのに役立ちます。

6 ディプロマティック版の処理

PrintFile_diplomatique.tex を $\text{\LaTeX}2_{\epsilon}$ で処理してみましょう。ここまでの作業を完璧にこなしていれば、処理は、通常、エラーなしに終わるはずです。万一、エラーが出た場合には、そのエラーがあなたのミスに起因しない場合があります。現在わかっているものとしては、略号と固有名の衝突があります。次の入力ファイルの例をみてください。

```
... \qu'A|MBORE...
```

上記のテキストはどこにも間違いはありません。 $\backslash\text{qu}'\text{A|}$ はディプロマティック版では $\backslash\text{qua|}$ に書き代わりますから略号マクロ名として何の問題もありません。しかし、別のところに問題があります。 $\backslash\text{qu}'\text{A|}$ の一部だけが固有名詞の一部になっているということです。ディプロマティック版では固有名詞には下線が引かれますが、こうした場合、Scriptorium は固有名詞の範囲を正しく確定できなくなります。実際、略号の一部に下線を引くというのは不可能です (\acute{q} と \grave{c} などの、どの部分が -e で、どの部分が -on かをいえる人はいないでしょう)。

エラーはディプロマティック版にしか起きません。けれども、クリティック版でも実は予想外のことが起っています。本来、略号は斜体で、固有名はスモールキャピタルで組まれなければなりません。今の場合、出力結果は $\llcorner \dots \text{qu}'\text{AMBORE} \llcorner$ となります。

したがって、略号と固有名が交錯する場合には、略号表記をあきらめて、

```
... qu'AMBORE...
```

と入力し、脚注などに断り書きを入れるのが最も無難なやり方でしょう。

7 注釈書類の作成と Scriptorium によるその処理

7.1 最初に

テキストにまつわる様々な注釈の大半は、写本からの転写が終わってから執筆することになります。けれども、テキストの欠落など、転写中に書き留めてしまった方がよい事柄もあります。また、読みに自信がない場合や、二通りの読みがあり得る場合などは、その場で注釈しておいた方が、後々の作業の役にも立ちます。説明の都合上、本文の作業を一通り終えてから、注釈作成に着手する、という手順になっていますが、実際には、最初から注釈書類を準備

するのが普通でしょう。とはいえ、Scriptorium での処理は、これまで述べた手順で進めるべきです。最初から注釈と本文を混ぜて処理を始めてしまうと、修正作業が非常に複雑なものになってしまいます。

なお、底本以外の写本の読みを巻末にまとめるのか、脚注に記載するのか、といったように、注釈をどのように構成するかは、校訂法の問題です。しかし、説明の都合上、まず、脚注の原稿作成法と組版を紹介します。

7.2 脚注の組版

Scriptorium では、簡単に複数の脚注領域を扱えるようになっていました。この点に関して、我々がお手本としたのは Claude Régnier の *La Prise d'Orange, chanson de geste de la fin du XII^e siècle* (Klincksieck, 1983, Paris) です。Régnier は脚注を上下三つの領域に分割し、最初の領域に底本の写本の様々な事象に関する記述や、修正の内容を記載しました。そして、二番目の領域に A 系列写本の異文、三番目の領域に B 系列写本の異文を収録しました。次のような具合です。

```
本文 1 行目
本文 2 行目
本文 3 行目
```

底本に関する記述

A 系列写本の異文

B 系列写本の異文

脚注は、内容が肥大化すると、ページによっては本文が数行になってしまうという欠点もあります。たとえば、Atkinson Jenkins の *Chanson de Roland, Oxford version* (D. C. Heath and Company, Boston, 1924) の 260-261 ページの本文は、それぞれ二行ずつしかありません。けれども、後注を参照する手間がなくなることを考えれば、そんな欠点など取るに足りません。今後は、このレイアウトが主流になるべきでしょう (もちろん、Scriptorium がこの方式を強制することはありません)。

Scriptorium を使えば、五つの脚注領域に異文を収録することができます。それに加えて、もう一つ、コメント用の脚注領域を利用することもできます。異文用の脚注領域は自由に改行できませんが、コメント用の脚注は $\text{\LaTeX}2_{\epsilon}$ の通常の脚注ですから、自由に改行することができます。

7.3 脚注書類の作製法と扱い方

まず、本文を書き込むファイルとは別に、注釈を書き留めるためのテキストファイルを **Scriptorium** で作成します（ファイル名は任意です）。できあがったテキストファイルを開くと、次のようになっています。なお、⇒はタブ文字を¶は改行文字を表しています。これらの文字は不可視文字といて、通常画面表示されません（テキストファイルを扱う作業の場合、これらの文字が表示できるソフトを使用した方がミスが少なくてすみます）。また、選んだ行数や書式により、数字の前の0の数は異なります。

```
/.0001./⇒¶
/.0002./⇒¶
/.0003./⇒¶
/.0004./⇒¶
...
```

/_と_/で挟まれた数字は詩行番号です。注釈と本文を対応づけるのは、本文の詩行番号です。したがって、本文の第一詩行に注釈を付したければ、/.0001./⇒の後に注釈を書き込みます。**Scriptorium** は詩行番号を手がかりに、注釈を適切な位置に流し込みます。ですから、次の点には十分に注意してください。

1. /_と_/は詩行を示すマークなので、他の用途には使えない。
2. /_と_/の間の数字を書き換えない。/_や_/を消さない。

なお詩行番号の書式を後から変更したり、行数を増やしたりすることは **Scriptorium** の画面上で簡単にできます。万一、作業の途中で本文詩行の脱落や重複に気づいた場合には、本文を修正するだけでなく、注釈書類も修正しなくてはなりません。

詩行の脱落に気づいた：脱落箇所に/_/_/を入れます。たとえば、第三行が脱落していたのであれば、次のようにします

```
/.0001./⇒¶
/.0002./⇒¶
/_/_/
/.0003./⇒¶
```

詩行の重複に気づいた：重複行を消します。たとえば、第二行が第一行と重複していたのであれば、次のようにします。

```
/.0001./⇒....¶
/.0003./⇒...¶
```

詩行の前後逆転に気づいた：正しい位置に戻します。たとえば、第二行と第三行が入れ替わっていたのであれば、次のようにします。

```
/.0001./⇒...¶
/.0003./⇒...¶
/.0002./⇒...¶
/.0004./⇒...¶
```

以上を行った後、**Scriptorium** で番号のつけ直しを行えば、行番号が正しく書きかわります。逆から言えば、理由もなく、上のような行為をしてはなりません。

Scriptorium で番号のつけ直しを行えば、あなたが入力したタブ文字や改行文字はすべて削除されます。

7.4 注釈書類の書式

後注も含めて、注釈書類は、基本的にあなたが入力した通りに、出力されます。が、いくつかの入力は特別な扱いを受けます。

7.4.1 改行とタブ

原則として、改行とタブは、通常の、**LaTeX2_ε** の規則にしたがって解釈されます。すなわち、タブを文の整形に利用することはできませんし、段落を改めるには改行文字を二つ連続して入力する必要があります。

ただし、異文用脚注の各領域内では、自由な改行はできません。したがって、注釈書類内であなたが改行文字をいくつ入力したとしても、それは入力の便宜をはかるためのものとして、無視されます。つまり、次のようになります。

入力

```
/.0001./⇒ «em\per[er]e» := corr. BED,
MORT, MOIG, SEG, BRAULT, DUF. || ¶
⇒ «magnes» [avec «es» effacé mais
lisible (≠ BED)] ¶
/.0002./⇒ «Espagne» [avec «e» effacé
mais lisible (≠ BED)] ...
```

出力 1

```
0001 «emper[er]e» := corr. BED, MORT, MOIG, SEG,
BRAULT, DUF. || «magnes» [avec «es» effacé mais li-
sible (≠ BED)] — 0002 «Espagne» [avec «e» effacé
mais lisible (≠ BED)] ...
```

異文を行ごとに改行することは、**Scriptorium** を設定すれば、可能です。その場合、先の入力原稿は次のように組版されます。

出力 2

```
0001 « emper[er]e » : = corr. BED, MORT, MOIG, SEG,
BRAULT, DUF. || « magnes » [avec « es » effacé mais li-
sible (≠ BED)]
0002 « Espagne » [avec « e » effacé mais lisible (≠
BED)] ...
```

7.4.2 L^AT_EX_{2_ε} コマンド

本文テキストファイルと異なり、注釈書類内では、L^AT_EX_{2_ε} コマンドがそのまま利用可能です（ただし、次に述べる、D 領域 [D= ディプロマティック] と C 領域 [C=クリティック] は例外です）。また、アクサン付きの文字やギュメなどは、巻末の使用可能文字一覧に含まれたものに関しては、Scriptorium が適切に変換しますので、\’e{, \og{ } といった入力法をとる必要はありません。

7.4.3 C 領域と D 領域

注釈内では、文字列を << ... >>（不等号の二重打ち）を挟むことで、その文字列を D 領域として、</ ... /> で挟むことで C 領域として、マーキングすることができます。これにより、略号コマンドが L^AT_EX_{2_ε} コマンドと区別され、適切に解釈されます。たとえば、em\perlere という略号コマンドを、C 領域に入れば、組版結果は「**emperere**」となります。一方、D 領域に入れば、「**empere**」となります。とはいえ、略号コマンドの解釈だけが問題となる訳ではないことにご注意ください。本文ファイルでの OLIVIER は、ディプロマティック版では「**oliuier**」となり、クリティック版では、「**OLIVIER**」となりますが、それと同じことが、C 領域と D 領域にもおこります。D 領域の文字列は、ディプロマティック版と全く同様に、C 領域の文字列はクリティック版と全く同様に組版されるのです（ただし、これらの領域の文字列は「&」で挟まれて組版されますし、C 領域はボールド体で組まれます。これらの設定は、スタイルファイルに手を加えることで、簡単に変更できます）。したがって、略号コマンドのあるなしに関わらず、ディプロマティック版からの引用は常に << ... >> で、クリティック版からの引用は </ ... /> でマーキングしておかねばなりません。また、略号コマンドが用いられたテキストを、C 領域にも D 領域にも入れなければ、エラーあるいは、予想外の組版が生じます。

7.4.4 使ってはいけない文字

1. \ は略号表記か L^AT_EX_{2_ε} コマンド以外の用途には使用できません。
2. | は L^AT_EX_{2_ε} 略号表記以外の用途には使えません（ただし、|| は || を出力するのに使用でき

ます。）

3. \$ は L^AT_EX_{2_ε} コマンド以外の用途には使えません。
4. <<>>, < / / > は領域のマーキング以外の用途には使えません。
5. /_と_/ は詩行番号マーキング以外の用途には使えません（既述）。

7.5 後注

前項、「注釈書類の書式」で述べたことからは、脚注だけでなく後注にもあてはまります。ただし、後注は、単なる地の文として組版されます。脚注のように、特別な組版を要しないので、何種類でも利用できます。ただし、多用するのは考えものです。たとえば、Gerard Brault は *The Song of Roland, an analytical edition, II, Oxford Text and English Translation* (The Pennsylvania State University Press, 1978, University Park and London) で、Notes to the Manuscript と Notes to the Text and Translation という二種類の後注を付しました。けれども、ある行に、それに関する注釈があるのかどうかを確かめるすべは、本文にはありません。しかも、本文に疑問を感じた時に、二種の注釈のいずれを参照すべきかも、必ずしも明瞭ではありません。結局、読者は本文を一行読むごとに、二種の後注をいちいち確認するか、後注のあるなしを本文にあらかじめ書き込むかの手間をかけるほかないのです。

以上を踏まえて、Scriptorium の後注には、ちょっとした工夫を施しました。読者が後注を見落とさないように、後注のある行には、本文の末尾に小さなマークが付きまゝ。けれども、繰り返しますが、後注を何種類も付すのは、読者泣かせです。

さて、後注の場合には、原稿の入力方法も若干異なります。脚注用の原稿と全く同じやりかたで入力しても構いませんが、後注用の原稿の方が自由度が高いのです。

注釈対象となる行番号を/_51_/などとするのは同じです。脚注原稿の場合と同様、行番号は/_0051_/などと 0 を付けても構いませんし、付ける方式と付けない方式が混在していても構いません。後で自動的に統一することができます。しかも、番号が連続している必要はありません。また、行番号だけで注釈の中身の無い場合には、無視されます。以上の自由の代償として、脚注原稿のように後から、番号を付け直すことはできません（番号の書式は変更できます）。

後注はコメント脚注と同様、\par{ } や改行文字の

二度打ちによって、自由に改行できます。

ただし、行番号の直前と直後の改行はいくつあっても全く無視されます（たとえ、一つも改行が入ってなくても、注釈同士を仕切る改行は自動入力されます。）。つまり、次のように組版されます。

入力
/_0049_/⇒第 49 詩行の注釈内容 ¶
/_0051_/ ¶
¶
第 51 詩行の注釈内容 ¶
¶
/_0068_/⇒第 68 詩行の ¶
注釈内容 ¶

出力
49..... 第 49 詩行の注釈内容
51..... 第 51 詩行の注釈内容
68..... 第 68 詩行の注釈内容

7.6 Scriptorium による注釈書類の処理

Scriptorium のウィンドウ上で、脚注書類、後注書類の指定ができます。指定した後は、必ず、設定ファイルを保存し直してください。

注釈書類は複数あったとしても、一度に、指定してしまわずに、ひとつ指定するごとに Compiler ボタンを押し、出来上がった、PrintFile.critique.tex と PrintFile.diplomatique.tex を L^AT_EX_{2_ε} で組版してください。おそらく、エラーがいくつも出るはずです。注釈書類の一つ一つについて、順を追ってミスを修正していかないと、どの書類に問題があったのかわからなくなり、非常にやっかいです。

注釈の D 領域で、本文では利用しなかった略号を使用している場合、次の手順で、新たに略号の定義を行わなければなりません。

■[手 順 1] TablesDesAbbreviations.tex を L^AT_EX_{2_ε} 処理します。すると、エラーが起ります。エラーの起った部分に、新しい略号が使用されているはずです。% を行頭につけてコメントアウトし、エラーなしに L^AT_EX_{2_ε} 処理が終わるまで、同じ手順を繰り返します。次にコメントアウトした、全ての略号の名前を控えておきます。

■[手 順 2] Abreivation.tex を result-files フォルダから取り除き、Scriptorium の処理を実行します。すると、新しい Abreivation.tex ができます。そこから、新しい略号の定義の部分だけをコピーし、古い Abreivation.tex の冒頭部にペーストします。

■[手 順 3] 新しい Abreivation.tex を捨て、古い Abreivation.tex を result-files フォルダに戻し、新しい略号の定義を行います。

慣れてくればこんな手間のかかることをしなくても、Abreivation.tex を直接手直してできるようになります。

7.7 使用可能文字

本文ファイル、脚注／後注ファイルとも、次の文字を利用することができます。これ以外の文字は L^AT_EX_{2_ε} コマンドを利用して入力することになります。ただし、L^AT_EX_{2_ε} コマンドを使用できるのは、脚注／後注ファイルの中だけです。

7.7.1 LowAscii 文字

shift 以外のモディフィケーション・キーを使わずに入力できる文字がこれにあたります。

```

0 1 2 3 4 5 6 7 8 9 a b c d e f g h i j
k l m n o p q r w t u v w x y z - [ ; : ]
' .

! " ' ( ) 0 = ' + * < > ? / A B C D E
F G H I J K L M N O P Q R S T U V W X Y Z

```

次の文字は LowAscii 文字ですが、L^AT_EX_{2_ε} コマンドに使用されるため、本文テキストファイル内では、所定の目的以外には使用できません。注釈書類ではエスケープすれば使用できます（バックスラッシュとチルドは数式モードでの入力が必要になります）。

```
# $ % & _ { } \ ^ ~ |
```

ただし、| は || とすれば、|| を出力するのに利用できます。また、本文ファイルの本文フィールドでは、スラッシュの二度打ち (//) が、詩行の分割に利用されることも忘れないでください。

7.7.2 アクサン記号付きアルファベ

```

é è ê ë ē ä à ä â ã ä í î ï ï ó ò
ô õ ö ø ú ù ü û ũ æ æ ç ñ ß ŷ
É È Ê Ë Ë Ä Å Ä Å Ä Ä Í Ì Î Ï Ó Ò
Ö Õ Ö Ö Ø Ú Ù Ü Ü Ů Ů Ů Ç Ñ Ÿ

```

7.7.3 記号類

```

| • ° ° ° © ® ™ £ ¶ ∞ ÷ ≤ ≥ “ ” ‘ ’
< > · « » ... ¿ ¡ − √ § ≠ ± π † ‡ ~ ≈ ◇ △

```